UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/068,466	02/05/2002	Sally Elaine Saffer	200302250-1	6758

7590 03/12/2008

IP Administration
Legal Dept., M/S 35
Hewlett-Packard Company
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
ORTIZ, BELIX M	
ART UNIT	**PAPER NUMBER**
2164	
MAIL DATE	**DELIVERY MODE**
03/12/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

80

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

MAR 12 2008

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/068,466
Filing Date: February 05, 2002
Appellant(s): SAFFER ET AL.

Robert Plotkin

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed November 24, 2006 appealing from the Office action mailed September 6, 2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

5,717,924	Kawai	7-1995

5,553,279	Goldring	10-1993
5,761,706	Kessler	11-1994
5,978,771	Vandivier, III	8-1997

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-4, 8-13, 29-35, 39-44, 62-63, and 66-67 are rejected under 35 U.S.C. 103(a) (Eff. Filing date of application 2/5/2002) as being unpatentable over Kawai (U.S. patent 5,717,924) (Eff. Filing date of application 7/7/1995) in view of Goldring (U.S. patent 5,553,279) (Eff. Filing date of application 10/8/1993).

Kawai and Goldring discloses:

As to claim 1, Kawai teaches an operational data store (see abstract; column 1, lines 8-10; and column 3, lines 29-33), comprising:

an insert table for storing new data (see figure 10A, character 320).

Kawai does not teach a history table for storing historical data; and

transfer logic for periodically transferring new data from the insert table to the history.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches a history table for storing historical data (see abstract; figure 2, character 46; column 1, lines 20-23; column 1, lines 46-54; column 2, lines 66-67; and column 3, lines 1-8 where in "history table" is read on "consistent change data table"); and

transfer logic for periodically transferring new data from the insert table to the history table (see column 6, lines 27-40; column 6, lines 62-67; column 7, lines 1-10; and column 8, lines 33-40).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Goldring because a history table for storing historical data; and

transfer logic for periodically transferring new data from the insert table to the history, would enable a operational data store because, "The change data noted above, including the change operation identifier that permits multi-generational snapshot copying without loss of change information, is placed in a Consistent.sub.-- Change.sub.-- Data table 46 by the Apply Processor 34. There is a Consistent.sub.-- Change.sub.-- Data table for each snapshot copy 38 of a user table 28. The Consistent.sub.-- Change.sub.-- Data table includes only updates that have been committed and is created by performing an SQL join operation on the Change.sub.-- Data 42 and UOW 44 tables. That is, after the Log Read Processor inserts information from the activity log 32 into the Change.sub.-- Data and UOW tables, the Apply Processor will read both tables and, based on records with the same log sequence number, will join information in the tables according to transaction updates that have committed since the time of the previous time stamp value, or most recent differential refresh operation", (see Goldring, column 6, lines 66-67 and column 7, lines 1-8).

As to claim 2, Kawai as modified teaches wherein the history table is partitioned (see column 2, lines 27-29).

As to claim 3, Kawai as modified teaches wherein the history table is partitioned by range (see Kawai, column 18, lines 45-46).

As to claim 4, Kawai as modified teaches wherein each partition is further sub-partitioned (see Kawai, Figures 2 and 5).

As to claim 8, Kawai as modified teaches the transfer logic comprising:

a secondary table (see Kawai, figure 10B, character 344);

fill logic for filling the secondary table with selected data from the insert table (see Kawai, figure 10B); and

secondary transfer logic for transferring the secondary table into the history table, the selected data thereby being transferred into the history table (see Kawai, figure 10B).

As to claim 9, Kawai as modified teaches wherein the history table has an indexing scheme, the secondary transfer logic (see, Kawai, figures 15A and 15B; column 10, line 17; and column 13, lines 9-19) further comprising:

indexing logic for applying the history table indexing scheme to the secondary table (see Kawai, figures 15A and 15B; column 10, line 17; and column 13, lines 9-19).

As to claim 10, Kawai as modified teaches wherein the indexing logic applies the history table indexing scheme to the secondary table prior to transferring the secondary table into the history table (see Kawai, figures 15A and 15B).

As to claim 11, Kawai as modified teaches the secondary transfer logic further comprising:

table logic for creating a new partition the history table, the new partition for swapping with the secondary table (see Kawai, table 2, lines 9).

As to claim 12, Kawai as modified teaches wherein the secondary transfer logic swaps the secondary table: and the new partition by exchanging respective pointers (see Kawai, column 10, line 14 and column 10, lines 25-30).

As to claim 13, Kawai, as modified teaches the operational data store further comprising:

a query engine for applying a database query to both the history table and the insert table (see Kawai, column 2, lines 19-21 and column 4, lines 29-42).

As to claim 29, Kawai as modified teaches wherein data from the insert table is transferred to the history table at regular intervals (see Kawai; column 16, lines 37-42 and column 16, lines 46-50).

As to claim 30, Kawai as modified teaches wherein the intervals are configurable (see Kawai, column 16, lines 37-52).

As to claim 31, Kawai teaches wherein the intervals are different for different tables (see Kawai, column 16, lines 38-42).

As to claim 32, Kawai teaches a method for maintaining an operational data store (see abstract; column 1, lines 8-10; and column 3, lines 29-33), comprising:

inserting new data into an insert table (see figure 10A, character 320).

Kawai does not teach periodically transferring data from the insert table to a history table.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches periodically transferring data from the insert table to a history table (see column 6, lines 27-40; column 6, lines 62-67 and column 7, lines 1-10 where in "history table" is read on "consistent change data table" and column 8, lines 33-40); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Goldring because periodically transferring data from the insert table to a history table, would enable a operational data store because, "The change data noted above, including the change operation identifier that permits multi-generational snapshot copying without loss of change information, is placed in a Consistent.sub.-- Change.sub.-- Data table 46 by the Apply Processor 34. There is a Consistent.sub.-- Change.sub.-- Data table for each snapshot copy 38 of a user table 28. The Consistent.sub.-- Change.sub.-- Data table includes only updates that have been committed and is created by performing an SQL join operation on the Change.sub.-- Data 42 and UOW 44 tables. That is, after the Log Read Processor inserts information from the activity log 32 into the Change.sub.-- Data and UOW tables, the Apply Processor will read both tables and, based on records with the same log sequence number, will join information in the tables according to transaction updates that have committed since the time of the previous time stamp value, or

most recent differential refresh operation", (see Goldring, column 6, lines 66-67 and column 7, lines 1-8).

As to claim 33, Kawai as modified teaches the method further comprising:

partitioning the history table (see Kawai, column 2, lines 27-29).

As to claim 34, Kawai as modified teaches wherein the history table is partitioned according to range (see Kawai, column 18, lines 45-46).

As to claim 35, Kawai as modified teaches the method further comprising:

sub-partitioning each partition (see Kawai, figures 2 and 5).

As to claim 39, Kawai as modified teaches the method further comprising:

creating a secondary table (see Kawai, figure 10B, character 344);

filling the secondary table with selected data from the insert table (see Kawai, figure 10B); and

transferring the secondary table into the history table, the selected data thereby being transferred into the history table (see Kawai, figure 10B).

As to claim 40, Kawai as modified teaches wherein the history table has an indexing scheme (see Kawai, figures 15A and 15B; column1o, line 17; and column 13, lines 9-19), the method further comprising:

applying the history table indexing scheme to the secondary table (see Kawai, figures 15A and 15B; column1o, line 17; and column 13, lines 9-19).

As to claim 41, Kawai as modified teaches wherein the history table indexing scheme is applied to the secondary table prior to transferring the secondary table into the history table (see Kawai, figures 15A and 15B).

As to claim 42, Kawai as modified teaches the method further comprising:

creating a new partition in the history table, wherein the secondary table is transferred by being swapped with the new partition (see Kawai, table 2, line 9).

As to claim 43, Kawai as modified teaches wherein the secondary table and new partition are swapped by exchanging respective pointers (see Kawai, column 10, line 14 and column 14, lines 25-30).

As to claim 44, Kawai as modified teaches the method further comprising:

applying a database query to both the history table and the insert table (see Kawai, column 2, lines 19-21 and column 4, lines 29-42).

As to claim 62, Kawai teaches an operational data store (see abstract; column 1, lines 8-10; and column 3, lines 29-33), comprising:

means for inserting new data into an insert table (see figure 10A, character 320);

and

means for applying a database query to both the history table and the insert table (see column 2, lines 19-21 and column 4, lines 29-42).

Kawai does not teach means for periodically transferring data from the insert table to a history table.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches means for periodically transferring data from the insert table to a history table (see column 6, lines 27-40; column 6, lines 62-67 and column 7, lines 1-10 where in "history table" is read on "consistent change data table" and column 8, lines 33-40);

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Goldring because means for periodically transferring data from the insert table to a history table, would enable a operational data store because, "The change data noted above, including the change operation identifier that permits multi-generational snapshot copying without loss of change information, is placed in a Consistent.sub.-- Change.sub.-- Data table 46 by the Apply Processor 34. There is a Consistent.sub.-- Change.sub.-- Data table for each snapshot copy 38 of a user table 28. The Consistent.sub.-- Change.sub.-- Data table includes only updates that have been committed and is created by performing an SQL join operation on the Change.sub.-- Data 42 and UOW 44 tables. That is, after the Log Read Processor inserts information from the activity log 32 into the Change.sub.-- Data and UOW tables, the Apply Processor will read both tables and, based on records with the same log sequence number, will join information in the tables according to transaction updates that have committed since the time of the previous time stamp value, or most

recent differential refresh operation", (see Goldring, column 6, lines 66-67 and column 7, lines 1-8).

As to claim 63, Kawai as modified teaches the operational data store further comprising:

means for batching new data (see Kawai column 2, lines 15-17); and

means for inserting the batched new data into the insert table with a single database access (see Kawai column 2, lines 34-36).

As to claim 66, Kawai teaches a computer program product for operating an operational data store, the computer program product comprising a computer usable medium having computer readable code thereon (see abstract; figure 1; and column 1, lines 6-8), including program code which:

inserts new data into an insert table (see figure 10A, character 320); and

applies a database query to both the; history table and the insert table (see column 2, lines 19-21 and column 4, lines 29-42).

Kawai does not teach periodically transfers data from the insert table to a history table.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches periodically transfers data from the insert table to a history table (see column 6, lines 27-40; column 6, lines 62-67 and column 7, lines 1-10 where in "history table" is read on "consistent change data table" and column 8, lines 33-40); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Goldring because periodically

transfers data from the insert table to a history table, would enable a operational data store because, "The change data noted above, including the change operation identifier that permits multi-generational snapshot copying without loss of change information, is placed in a Consistent.sub.-- Change.sub.-- Data table 46 by the Apply Processor 34. There is a Consistent.sub.-- Change.sub.-- Data table for each snapshot copy 38 of a user table 28. The Consistent.sub.-- Change.sub.-- Data table includes only updates that have been committed and is created by performing an SQL join operation on the Change.sub.-- Data 42 and UOW 44 tables. That is, after the Log Read Processor inserts information from the activity log 32 into the Change.sub.-- Data and UOW tables, the Apply Processor will read both tables and, based on records with the same log sequence number, will join information in the tables according to transaction updates that have committed since the time of the previous time stamp value, or most recent differential refresh operation", (see Goldring, column 6, lines 66-67 and column 7, lines 1-8).

As to claim 67, Kawai as modified teaches wherein the program code further:

batches new data (see Kawai column 2, lines 15-17); and

inserts the batched new data into the insert table with a single database access (see Kawai column 2, lines 34-36).

Claims 14-22 and 45-53 are rejected under 35 U.S.C. 103(a) (Eff. Filing date of application 2/5/2002) as being unpatentable over Kawai (U.S. patent 5,717,924) (Eff. Filing date of application 7/7/1995) in view of Goldring (U.S. patent 5,553,279) (Eff. Filing date of

application 10/8/1993) as applied to claims 1-4, 8-13, 29-35, 39-44, 62-63 above, and 66-67 in view of Kessler et al. (U.S. patent 5,761,706).

As to claim 14, Kawai teaches the operational data store further comprising:

an aggregator for batching the accumulated data and transferring the batched data into the insert table with a single database access (see column 2, lines 34-36).

Kessler et al. teaches stream buffers for high-performance computer memory system (see abstract), in which he teaches an aggregation buffer for accumulating new data (see abstract and column 1, lines 6-8).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Kessler et al., because an aggregation buffer for accumulating new data, would enable the method because, it is advantageous to replace or supplement the secondary cache with buffers. Buffers require much less hardware to implement, yet can provide performance similar to a large secondary cache. "Method and apparatus for a filtered stream buffer coupled to a memory and a processor, and operating to prefetch data from the memory. The filtered stream buffer includes a cache block storage area and a filter controller. The filter controller determines whether a pattern of references has a predetermined relationship, and if so, prefetches stream data into the cache block storage area ", (see Kessler et al., abstract).

As to claim 15, Kawai, as modified teaches wherein the aggregator transfers a batch of new data into the insert table when the batch surpasses a maximum size (see Kessler et al., column 11, lines 30-34).

As to claim 16, Kawai, as modified teaches wherein batch size is measured according to a number data bytes (see Kessler et al., figure 11 and column 14, lines 29-30).

As to claim 17, Kawai, as modified teaches wherein batch size is measured according to a number of records (see Kessler et al., figures 9 and 11).

As to claim 18, Kawai, as modified teaches wherein the maximum size is configurable (see Kessler et al., figures 11 and 12 and column 16, lines 30-35).

As to claim 19, Kawai, as modified teaches wherein the aggregator transfers batches of new data into the insert table at regular intervals, defined by a given period (see Kessler et al., column 1, lines 18-20).

As to claim 20, Kawai, as modified teaches wherein the period is configurable (see Kessler et al., column 1, lines 11-20).

As to claim 21, Kawai, as modified teaches wherein the aggregator transfers batches of new data into the insert table when the aggregation buffer surpasses a given maximum buffer size (see Kessler et al., column 6, lines 51-53).

As to claim 22, Kawai, as modified teaches wherein the maximum buffer size is configurable (see Kessler et al., figures 11 and 12 and column 16, lines 30-35).

As to claim 45, Kawai teaches the method further comprising:

inserting the batched new data into the insert table with a single database access (see column 2, lines 34-36).

Kessler et al. teaches stream buffers for high-performance computer memory system (see abstract), in which he teaches aggregating new data into batches (see abstract and column 14, lines 55-57).

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Kessler et al., because aggregating new data into batches, would enable the method because, batching the data makes easier and fast the transfer of the data to the corresponding table and is going to be just a single access to the memory.

As to claim 46, Kawai, as modified teaches wherein a batch of new data is transferred into the insert table when the batch surpasses a maximum size (see Kessler et al., column 11, lines 30-34).

As to claim 47, Kawai, as modified teaches wherein batch size is measured according to a number data bytes (see Kessler et al., figure 11 and column 14, lines 29-30).

As to claim 48, Kawai, as modified teaches wherein batch size is measured according to a number of records (see Kessler et al., figures 9 and 11).

As to claim 49, Kawai, as modified teaches wherein the maximum size is configurable (see Kessler et al., figures 11 and 12 and column 16, lines 30-35).

As to claim 50, Kawai, as modified teaches wherein batches of new data are transferred into the insert table at regular intervals, defined by a given period (see Kessler et al., column 1, lines 18-20).

As to claim 51, Kawai, as modified teaches wherein the period is configurable (see Kessler et al., column 1, lines 11-20).

As to claim 52, Kawai, as modified teaches the method further comprising:

- aggregating the batches of new data in an aggregation buffer, wherein the batches are transferred into the insert table when the aggregation buffer surpasses a given maximum buffer size (see Kessler et al., column 6, lines 51-53).

As to claim 53, Kawai, as modified teaches wherein the maximum buffer size is

configurable (see Kessler et al., figures 11 and 12 and column 16, lines 30-35).

Claims 60-61, 65, and 69 are rejected under 35 U.S.C. 103(a) (Eff. Filing date of application 2/5/2002) as being unpatentable over Kawai (U.S. patent 5,717,924) (Eff. Filing date of application 7/7/1995) in view of Vandivier, III (U.S. patent 5,978,771) (Eff. Filing date of application 8/8/1997).

As to claims 60 and 65, Kawai teaches a method for operating an operational data store (see abstract; column 1, lines 8-10; and column 3, lines 29-33), comprising:

creating a partitioned temporary table (see figure 10B, character 332);

filling the temporary table with data from an insert table (see figure 10B, character 336);

exchanging the temporary table with the new partition (see figure 10B, character 348); and

receiving a query and applying the query to both the history table and the insert table (see column 2, lines 19-21 and column 4, lines 29-42).

Kawai does not teach creating a new partition in a composite-partitioned history table.

Vandivier, III teaches method for tracking natural resources on a resource allocation system (see abstract), in which he teaches creating a new partition in a composite-partitioned history table (see column 11, lines 20-27 and column 12, lines 19-23); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Vandivier, III because creating a new partition in a composite-partitioned history table, would enable a operational data store

because, "User can add a new partition set 603 and define the number of constituent components corresponding to this new category. Default partition ratios for each component are entered such that the sum of the ratios each 100 percent. A user can modify 604 any constituent component of a partition set by editing partition ratios for each component. A user can also delete 605 any partition set category and its corresponding partition set database records from selected partition set", (see Vandivier, III, column 11, lines 25-33).

As to claim 61, Kawai as modified teaches the method further comprising:

creating a new partition in the insert table based on values from an existing partition (see Kawai figure 7); and

dropping the existing partition (see Kawai figure 10B, character 340).

As to claim 69, Kawai teaches a computer program product for operating an operational data store (see abstract; column 1, lines 8-10; and column 3, lines 29-33), the computer program product comprising a computer usable medium having computer readable code thereon (see abstract; figure 1; and column 1, lines 6-8), including program code which:

creates a partitioned temporary table (see figure 10B, character 332);

fills the temporary table with data from an insert table (see figure 10B, character 336);

exchanges the temporary table with the new partition (see figure 10B, character 348); and

receives queries and applies the queries to both the history table and the insert table (see column 2, lines 19-21 and column 4, lines 29-42).

Kawai does not teach creates a new partition in a composite-partitioned history table.

Vandivier, III teaches method for tracking natural resources on a resource allocation system (see abstract), in which he teaches creating a new partition in a composite-partitioned history table (see column 11, lines 20-27 and column 12, lines 19-23); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Vandivier, III because creating a new partition in a composite-partitioned history table, would enable a operational data store because, "User can add a new partition set 603 and define the number of constituent components corresponding to this new category. Default partition ratios for each component are entered such that the sum of the ratios each 100 percent. A user can modify 604 any constituent component of a partition set by editing partition ratios for each component. A user can also delete 605 any partition set category and its corresponding partition set database records from selected partition set", (see Vandivier, III, column 11, lines 25-33).

Claims 70-73 are rejected under 35 U.S.C. 103(a) (Eff. Filing date of application 2/5/2002) as being unpatentable over Battas et al. (U.S. patent 6,757,706) (Eff. Filing date of application 9/7/2001) in view of Goldring (U.S. patent 5,553,279) (Eff. Filing date of application 10/8/1993).

As to claim 70, Battas et al. teaches a system for producing a desired level of service in a mixed workload environment (see column 20, lines 31-34), comprising:

a high-speed insert operational data store (ODS) (see column 15, lines 34-37);

and

an aggregator for accumulating transactions into batches and inserting each of the batches into the ODS using a single database transaction per batch (see column 15, lines 41-46).

Battas et al. does not teach a throttler for throttling selected transactions to the ODS.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches a throttler for throttling selected transactions to the ODS (see column 7, lines 10-15); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Vandivier, III because a throttler for throttling selected transactions to the ODS, would enable a operational data store because, "As a result, the rows in the Consistent.sub.-- Change.sub.-- Data table will provide a listing of the update transactions in the order in which they were committed, sequence information used to order conflicting updates within a single transaction, and operational information used to specify whether a change was an insert, update, or delete operation", (see Goldring, column 7, lines 10-15).

As to claim 71, Battas et al. as modified teaches wherein the mixed workload environment includes at least two of archiving, OL.TP queries, DSS queries, high-speed inserts, backup processes and extract/translate/load transactions (see Battas et al. figure 8; column 2, lines 50-54; and column 20, lines 28-33).

As to claim 72, Battas et al. teaches a method for producing a desired level of service in a mixed workload environment (see column 20, lines 31-34), comprising:

insert transactions into an operational data store (ODS) at a high-speed (see column 15, lines 34-37);

accumulating transactions into batches (see column 15, lines 41-46); and

inserting each of the batches into the ODS using a single database transaction per batch (see column 15, lines 41-46).

Battas et al. does not teach throttling selected transactions to the ODS.

Goldring teaches lossless distribution of time series data in a relational data base network (see abstract), in which he teaches throttling selected transactions to the ODS (see column 7, lines 10-15); and

It would have been obvious to a person having ordinary skill in the art at the time the invention was made to have modified Kawai by the teaching of Vandivier, III because throttling selected transactions to the ODS, would enable a operational data store because, "As a result, the rows in the Consistent.sub.-- Change.sub.-- Data table will provide a listing of the update transactions in the order in which they were committed, sequence information used to order conflicting updates within a single transaction, and operational information used to specify whether a change was an insert, update, or delete operation", (see Goldring, column 7, lines 10-15).

As to claim 73, Battas et al. as modified teaches wherein the mixed workload environment includes at least two of archiving, OLTP queries, DSS queries, high-speed inserts, backup processes and extract/translate/load transactions (see Battas et al. figure 8; column 2, lines 50-54; and column 20, lines 28-33).

(10) Response to Argument

Firstly, Appellant argues that Goldring fail to disclose or suggest transferring data from an insert table to a history table.

In response, Examiner maintains that Goldring teaches "transferring data from an insert table to a history table" because on the Spec. paragraph [0043] the inventor explain than "transferring" is moving data from on table to another and does not mention transferring at any moment. Goldring on col. 6, lines 27-41 teach transferring the information from two system tables.

Secondly, Appellant argues that Vandivier, III failed to disclose or suggest creating a new partition in a composite-partitioned history table.

In response, Examiner maintains that Vandivier, III teaches "FIG. 2M illustrates how a partition table is updated 601. A user selects a partition set 602 from a partition set currently associated with a selected resource record in the resource allocation table, the resource stockpile table, or for generic resources available at selected location-levels. User can add a new partition set 603 and define the number of constituent components corresponding to this new category. Default partition ratios for each component are entered such that the sum of the ratios each 100 percent. A user can modify 604 any constituent component of a partition set by editing partition ratios for each component. A user can also delete 605 any partition set category and its corresponding partition set database records from selected partition set. A user is returned to the

tabbed page where user requested 606 to update the partition table", (see Vandivier, III, col. 11, lines 20-35).

"A new partition table of partition sets is generated for each partition record transferred into the allocation identity buffer. This allows future partitioning of newly entered records. If user elected 512 to exit partitioning operation without saving partitioned records, all temporary database buffers associated with such partitioning are released", (see Vandivier, III, col. 12, lines 19-23).

Thirdly, Appellant argues that Goldring failed to disclose or suggest a throttler for throttling selected transactions to the ODS.

In response, Examiner maintains that Goldring teaches "As a result, the rows in the Consistent.sub.-- Change.sub.-- Data table will provide a listing of the update transactions in the order in which they were committed, sequence information used to order conflicting updates within a single transaction, and operational information used to specify whether a change was an insert, update, or delete operation", (see Goldring, column 7, lines 10-15). Throttler is use to slowdown the process and let the first on the hierarchy to by first and Goldring teaches list the update order in which they were committed.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Belix M Ortiz
Art unit 2164

November 28, 2007

Conferees:

C. Rones
CHARLES RONES
SUPERVISORY PATENT EXAMINER

Charles L. Rones, SPE

Eddie Lee, SPE

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER

Robert Plotkin, P.C.

91 Main street, Suite 204

Concord, Ma 01742-2527